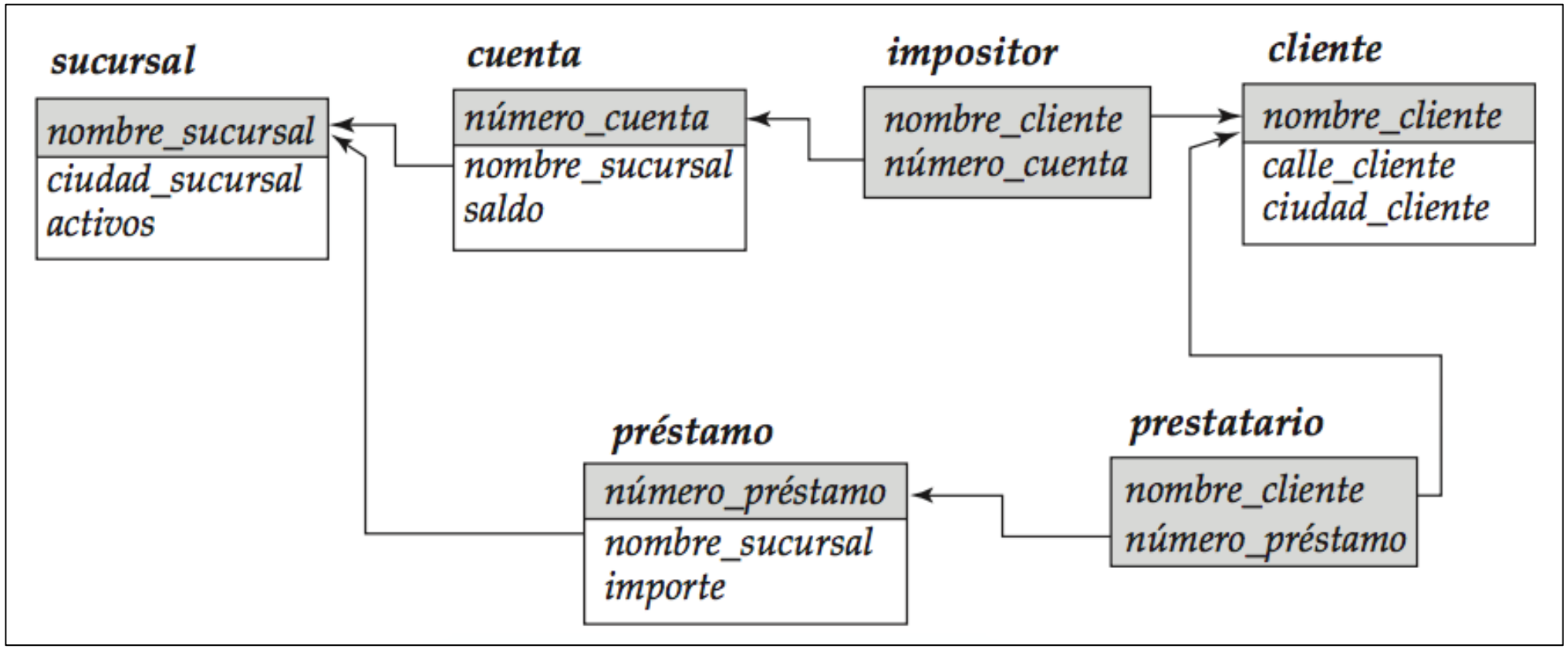
Bases de Datos

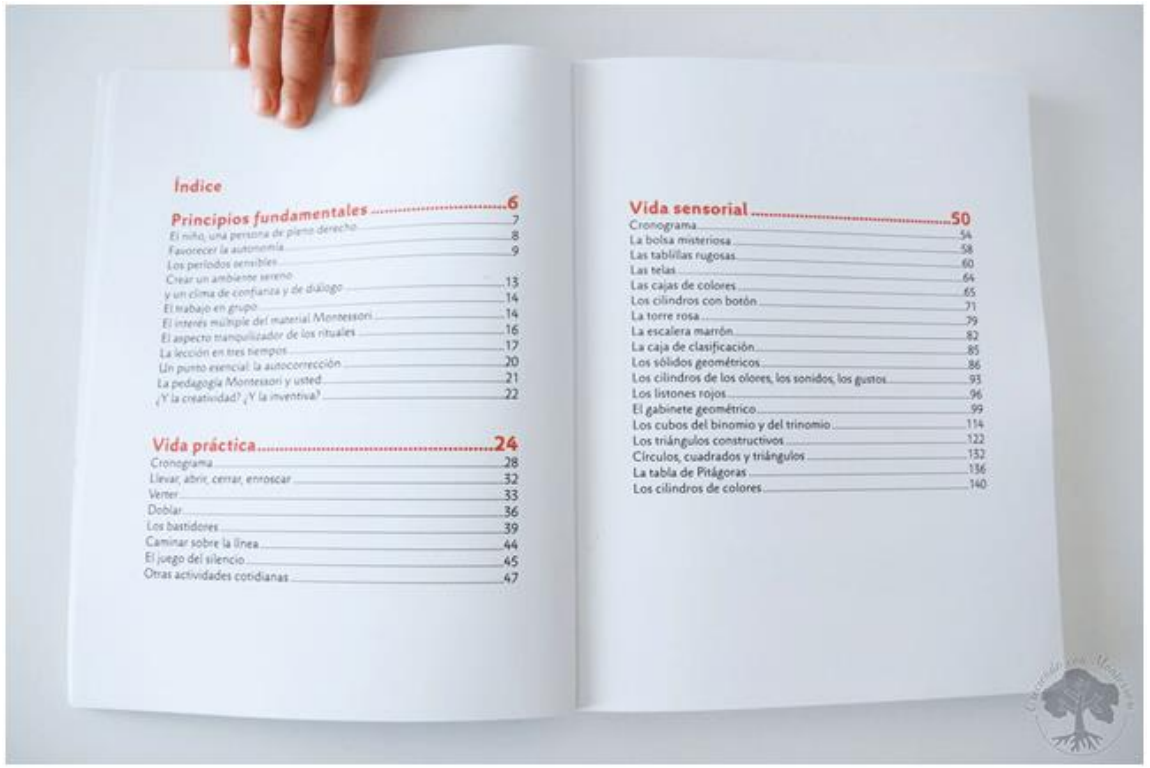
- ÍNDICES
- VISTAS
- USUARIOS, AUTENTICACIÓN Y AUTORIZACIÓN
- TRANSACCIONES

Lenguaje de Consultas (SQL)

- Índices
- Vistas
- Usuarios, autenticación y autorización
- Transacciones

Base de datos ejemplo. Banco





Lenguaje de Consultas (SQL)

- **Índices**

- Un índice para un archivo del sistema funciona como el índice de un libro. Si se va a buscar un tema (especificado por una palabra o una frase) se puede buscar en el índice al final del libro, encontrar las páginas en las que aparece y después leerlas para encontrar la información buscada.
- Las palabras del índice están ordenadas alfabéticamente, lo cual facilita la búsqueda. Además, el índice es mucho más pequeño que el libro, con lo que se reduce aún más el esfuerzo necesario para encontrar las palabras en cuestión.

Lenguaje de Consultas (SQL)

Índices

- Los índices de los sistemas de bases de datos juegan el mismo papel que los índices de los libros en las bibliotecas.
- Por ejemplo, en la base de datos banco, para recuperar un registro *cuenta* dado su número de cuenta, el sistema de bases de datos buscaría en un índice para encontrar el bloque de disco en que se localice el registro correspondiente, y entonces extraería ese bloque de disco para obtener el registro *cuenta*.

Índices

Sirven para agilizar la obtención de registros en respuesta a ciertas condiciones de búsqueda.

Hay unos tipos de índices, los denominados caminos secundarios de acceso, que ofrecen caminos alternativos de búsqueda, para localizar eficientemente los registros con base en los **campos de indización**.

Lenguaje de Consultas (SQL)

Índices

- Un índice es una estructura física de acceso que se especifica con base en uno o más atributos de un archivo.
- El atributo o atributos sobre los cuales se crea un índice se denominan atributos de indización.
- Los índices hacen más eficiente el acceso a tuplas con base en condiciones en las que intervienen sus atributos de indización.

Lenguaje de Consultas (SQL)

Índices

- La ejecución de una consulta tardará menos si algunos de los atributos implicados en las condiciones de la consulta están indizados.
- La mejoría puede ser enorme en el caso de consultas de relaciones grandes, si están indizados los atributos que intervienen en las condiciones de selección y de reunión de una consulta, el tiempo de ejecución de ésta se reduce considerablemente.

EMPLEADO

NOMBREP	INIC	APELLIDO	NSS	FECHAN	DIRECCIÓN	SEXO	SALARIO	NSSSUPER	ND

DEPARTAMENTO

NOMBRED	NÚMEROD	NSSGTE	FECHAINICGTE

LUGARES_DEPTOS

NÚMEROD	LUGARD

PROYECTO

NOMBREP	NÚMEROP	LUGARP	NÚMD

TRABAJA–EN

NSSE	NÚMP	HORAS

DEPENDIENTE

NSSE	NOMBRE–DEPENDIENTE	SEXO	FECHAN	PARENTESCO

Lenguaje de Consultas (SQL)

Índices

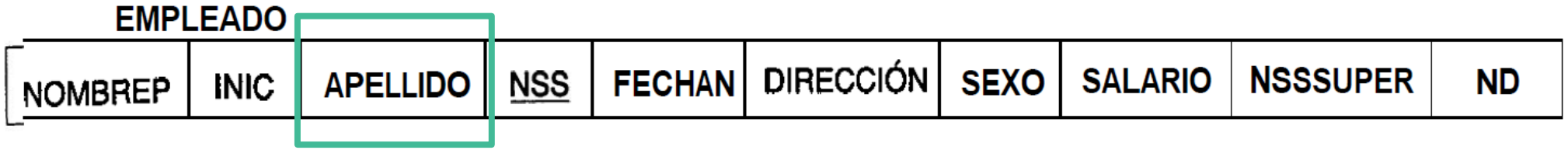
- Los índices se pueden crear y desechar dinámicamente.
- El comando CREATE INDEX sirve para especificar un índice, el cual recibe un nombre que se usará para desecharlo cuando ya no se necesite.

Lenguaje de Consultas (SQL)

Índices

Por ejemplo, para crear un índice sobre el atributo APELLIDO de la relación base EMPLEADO, podemos emitir el comando siguiente

El índice está en orden ascendente de los valores del atributo de indización.




```
CREATE   INDEX   ÍNDICE_APELLIDO
         ON      EMPLEADO (APELLIDO );
```

Lenguaje de Consultas (SQL)

Índices

También podemos crear un índice sobre una combinación de atributos. Por ejemplo, si queremos crear un índice basado en la combinación de NOMBREP, INIC y APELLIDO.

APELLIDO se encontrará en orden ascendente y NOMBREP en orden descendente dentro del mismo valor de APELLIDO:

EMPLEADO

NOMBREP	INIC	APELLIDO	NSS	FECHAN	DIRECCIÓN	SEXO	SALARIO	NSSSUPER	ND

```
CREATE   INDEX   ÍNDICE_NOMBRES
         ON      EMPLEADO (APELLIDO ASC, NOMBREP DESC, INIC );
```

Lenguaje de Consultas (SQL)

Índices

Índice de agrupamiento

Si los registros de un archivo están ordenados físicamente según un campo no clave que no tiene un valor distinto para cada registro, dicho campo se denomina campo de agrupamiento.

Y se puede crear un tipo diferente de índice, llamado índice de agrupamiento, para acelerar la obtención de registros que tienen el mismo valor en el campo de agrupamiento.

Lenguaje de Consultas (SQL)

Índices

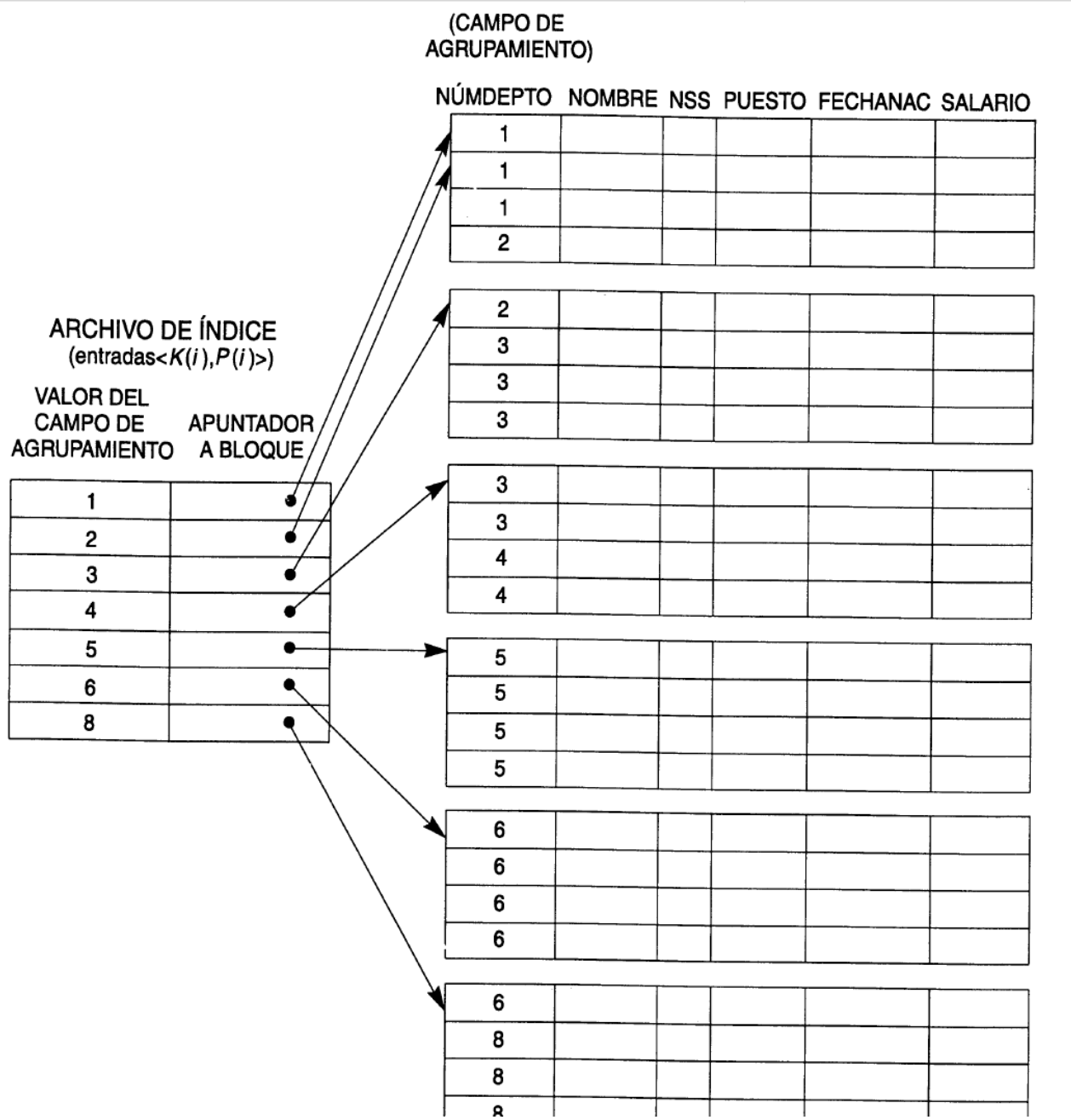
Un índice de agrupamiento es un archivo ordenado con dos campos;

el primero es del mismo tipo que el campo de agrupamiento del archivo de datos,

y el segundo es un apuntador a bloque.

Hay una entrada en el índice de agrupamiento por cada valor distinto del campo de agrupamiento, y contiene

el valor y un apuntador al primer bloque del archivo de datos que tiene un registro con ese valor en el campo de agrupamiento.



Lenguaje de Consultas (SQL)

Índices

- Al crear un índice es posible especificar si se trata o no de un índice de agrupamiento.
- Se usa la palabra reservada CLUSTER (grupo) al final del comando CREATE INDEX.
- Por ejemplo, si queremos que los registros EMPLEADO estén indizados y agrupados por número de departamento, crearemos un índice de agrupamiento sobre ND:

```
CREATE   INDEX      ÍNDICE_ND
         ON         EMPLEADO ( ND )
         CLUSTER;
```

Lenguaje de Consultas (SQL)

Índices

- El comando DROP INDEX sirve para desechar un índice.
- Los índices se desechan porque su mantenimiento resulta costoso cada vez que se actualiza la relación base y requieren almacenamiento adicional.
- Por ello, si ya no se emitirán consultas en las que interviene un atributo indizado, conviene desechar ese índice.

```
DROP INDEX    ÍNDICE_ND;
```

Lenguaje de Consultas (SQL)

Índices

```
MariaDB [banco]> create index indice_cuenta on cuenta(numero_cuenta);
Query OK, 0 rows affected (0.110 sec)
Records: 0  Duplicates: 0  Warnings: 0
```

```
MariaDB [banco]> drop index indice_cuenta on cuenta;
Query OK, 0 rows affected (0.212 sec)
Records: 0  Duplicates: 0  Warnings: 0
```

Lenguaje de Consultas (SQL)

Vistas

- En el mundo real, existen ocasiones en las que no resulta deseable que todos los usuarios vean el modelo lógico completo de una base de datos.
- Las consideraciones de seguridad pueden exigir que se oculten ciertos datos a los usuarios.
- Por ejemplo, una persona necesita saber el número de préstamo y el nombre de la sucursal de un cliente, pero no necesita ver el importe de ese préstamo. Esa persona debería ver una relación de la siguiente manera:

```
select nombre_cliente, prestatario.número_préstamo, nombre_sucursal
from prestatario, préstamo
where prestatario.número_préstamo = préstamo.número_préstamo
```

Lenguaje de Consultas (SQL)

Vistas

- Aparte de las consideraciones de seguridad, puede que se desee crear un conjunto personalizado de relaciones que se adapte mejor a la intuición de un usuario determinado que el modelo lógico.
- Puede que a un usuario del departamento de publicidad, por ejemplo, le guste ver una relación que consista en los clientes que tienen o bien cuenta abierta o bien préstamo concedido en el banco y las sucursales con las que trabajan. La relación que se crearía para ese empleado es la siguiente:

```
(select nombre_sucursal, nombre_cliente
from impositor, cuenta
where impositor.número_cuenta = cuenta.número_cuenta)
 union
(select nombre_sucursal, nombre_cliente
from prestatario, préstamo
where prestatario.número_préstamo = préstamo.número_préstamo)
```

Lenguaje de Consultas (SQL)

Vistas

Las relaciones que no forman parte del modelo lógico pero se hacen visibles a los usuarios como relaciones virtuales se denominan **vistas**.

Es posible definir un gran número de vistas de cualquier conjunto dado de relaciones reales.

Lenguaje de Consultas (SQL)

Vistas

Las vistas en SQL se definen mediante la instrucción **create view**. Para definir una vista hay que darle un nombre e indicar la consulta que la va a calcular. La forma de la instrucción **create view** es:

create view v **as** <expresión de consulta>

donde <expresión de consulta> es cualquier expresión legal de consulta. El nombre de la vista se representa mediante *v*.

Lenguaje de Consultas (SQL)

Vistas

Considérese la vista con las sucursales y sus clientes llamada *todos_los_clientes*.

Esta vista se define de la manera siguiente:

```
create view todos_los_clientes as
      (select nombre_sucursal, nombre_cliente
       from impositor, cuenta
       where impositor.número_cuenta = cuenta.número_cuenta)
     union
      (select nombre_sucursal, nombre_cliente
       from prestatario, préstamo
       where prestatario.número_préstamo = préstamo.número_préstamo)
```

Lenguaje de Consultas (SQL)

Vistas

- Una vez definida la vista se puede utilizar su nombre para hacer referencia a la relación virtual que genera.
- Utilizando la vista *todos_los_clientes* se puede determinar el nombre de todos los clientes de la sucursal de Navacerrada escribiendo:

```
select nombre_cliente
from todos_los_clientes
where nombre_sucursal = 'Navacerrada'
```

Lenguaje de Consultas (SQL)

Vistas

- Actividad. Elabora las siguientes vistas en la base de datos Banco, copia las imágenes con los trabajado en la consola de MariaDB y pégalas en un PDF:

1) La vista **todos_los_clientes** contendrá los clientes del banco con un préstamo, una cuenta o ambos:

```
create view todos_los_clientes as
    (select nombre_sucursal, nombre_cliente
     from impositor, cuenta
     where impositor.número_cuenta = cuenta.número_cuenta)
  union
    (select nombre_sucursal, nombre_cliente
     from prestatario, préstamo
     where prestatario.número_préstamo = préstamo.número_préstamo)
```

Lenguaje de Consultas (SQL)

Vistas

- Actividad.

1) La vista todos_los_clientes contendrá los clientes del banco con un préstamo, una cuenta o ambos:

```
MariaDB [banco]> select * from todos_los_clientes;
+-----------------+----------------+
| nombre_sucursal | nombre_cliente |
+-----------------+----------------+
| Centro          | Gonz├ílez      |
| Navacerrada     | L├│pez         |
| Galapagar       | Gonz├ílez      |
| Becerril        | G├│mez         |
| Galapagar       | Santos         |
| Moralzarzal     | Rup├®rez       |
| Collado Mediano | Abril          |
| Navacerrada     | Fern?ndez      |
| Collado Mediano | G?mez          |
| Moralzarzal     | G?mez          |
| Navacerrada     | L?pez          |
| Becerril        | P?rez          |
| Centro          | Santos         |
| Centro          | Valdivieso     |
+-----------------+----------------+
14 rows in set (0.465 sec)
```

Lenguaje de Consultas (SQL)

Vistas

- Actividad.
- Elabora las siguientes vistas en la base de datos Banco:

2) La vista **sucursales_cuenta** con el nombre de todas las sucursales con clientes que tienen una cuenta abierta en el banco.

```
MariaDB [banco]> create view sucursales_cuenta as
    -> select nombre_sucursal from cuenta natural join impositor;
Query OK, 0 rows affected (0.011 sec)

MariaDB [banco]> select * from sucursales_cuenta;
+-----------------+
| nombre_sucursal |
+-----------------+
| Collado Mediano |
| Becerril        |
| Centro          |
| Galapagar       |
| Navacerrada     |
| Moralzarzal     |
| Galapagar       |
+-----------------+
7 rows in set (0.006 sec)

MariaDB [banco]>
```

Lenguaje de Consultas (SQL)

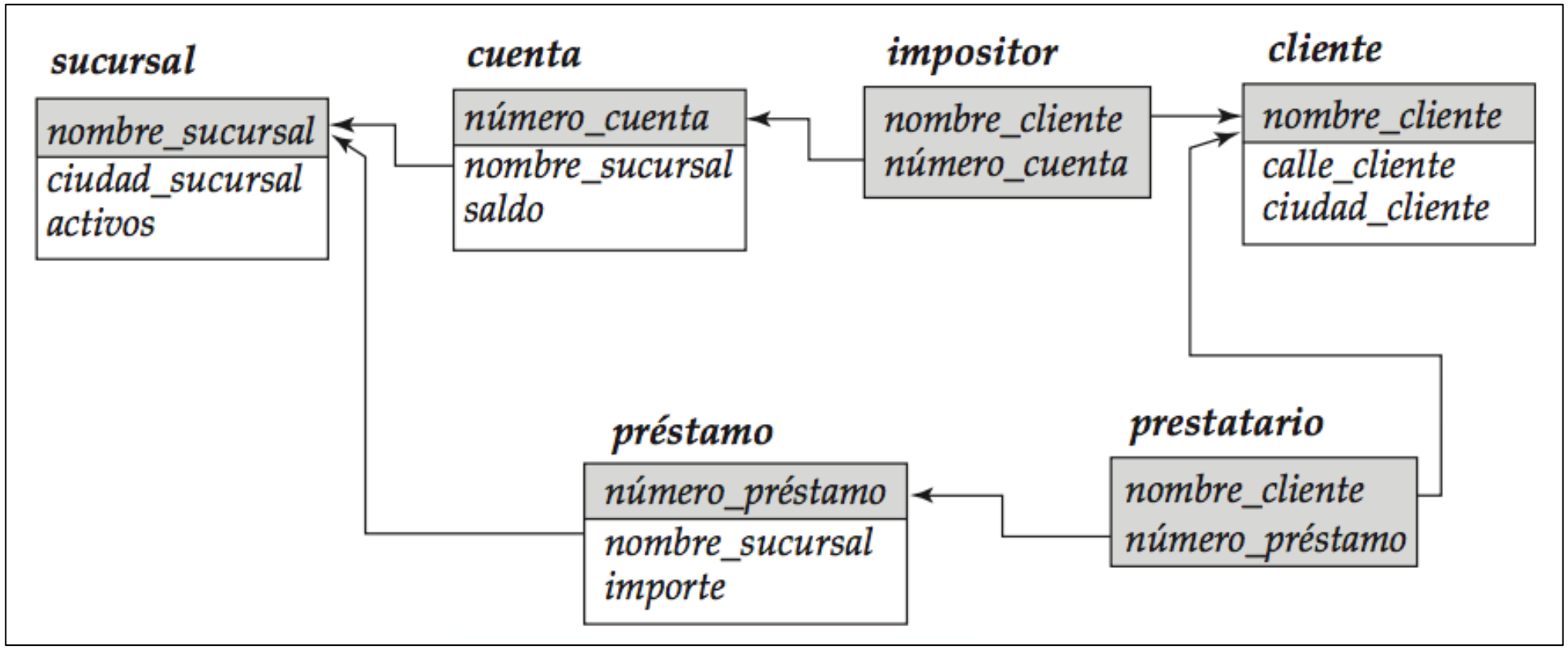
Vistas

- Actividad.
- Elabora las siguientes vistas en la base de datos Banco:

2) La vista **sucursales_cuenta** con el nombre de todas las sucursales con clientes que tienen una cuenta abierta en el banco.

```
MariaDB [banco]> create view sucursales_cuenta as
    -> select nombre_sucursal from cuenta natural join impositor;
Query OK, 0 rows affected (0.011 sec)

MariaDB [banco]> select * from sucursales_cuenta;
+-----------------+
| nombre_sucursal |
+-----------------+
| Collado Mediano |
| Becerril        |
| Centro          |
| Galapagar       |
| Navacerrada     |
| Moralzarzal     |
| Galapagar       |
+-----------------+
7 rows in set (0.006 sec)

MariaDB [banco]>
```

Base de datos ejemplo. Banco



Lenguaje de Consultas (SQL)

Vistas

- Actividad.
- Elabora las siguientes vistas en la base de datos Banco:

3) La vista **sucursales_saldo** con el nombre de todas las sucursales y el promedio del saldo de todas las cuentas por sucursal.

```
MariaDB [banco]> create view sucursales_saldo as
    -> select nombre_sucursal, avg(saldo) from cuenta group by nombre_sucursal;
Query OK, 0 rows affected (0.011 sec)

MariaDB [banco]> select * from sucursales_saldo;
+-----------------+------------+
| nombre_sucursal | avg(saldo) |
+-----------------+------------+
| Becerril        |   700.0000 |
| Centro          |   500.0000 |
| Collado Mediano |   350.0000 |
| Galapagar       |   825.0000 |
| Moralzarzal     |   700.0000 |
| Navacerrada     |   400.0000 |
+-----------------+------------+
6 rows in set (0.004 sec)
```

Lenguaje de Consultas (SQL)

Vistas

- Actividad.
- Elabora las siguientes vistas en la base de datos Banco:

4) La vista **sucursales_impositor** con el nombre de todas las sucursales y el número de impositores por cada una de ellas.

```
MariaDB [banco]> create view sucursales_impositor as
    -> select nombre_sucursal, count(nombre_cliente) from impositor natural join cuenta group by nombre_sucursal;
Query OK, 0 rows affected (0.018 sec)
```

Lenguaje de Consultas (SQL)

Vistas

- Actividad.
- Elabora las siguientes vistas en la base de datos Banco:

4) La vista **sucursales_impositor** con el nombre de todas las sucursales y el número de impositores por cada una de ellas.

```
MariaDB [banco]> select * from  sucursales_impositor;
+-----------------+-------------------------+
| nombre_sucursal | count(nombre_cliente) |
+-----------------+-------------------------+
| Becerril        |                       1 |
| Centro          |                       1 |
| Collado Mediano |                       1 |
| Galapagar       |                       2 |
| Moralzarzal     |                       1 |
| Navacerrada     |                       1 |
+-----------------+-------------------------+
6 rows in set (0.004 sec)
```

Lenguaje de Consultas (SQL)

Vistas

- Actividad.
- Elabora las siguientes vistas en la base de datos Banco:

5) La vista **sucursales_prestatario** con el nombre de todas las sucursales y el número de prestatarios por cada una de ellas.

```
MariaDB [banco]> create view sucursales_prestatario as
    -> select nombre_sucursal, count(nombre_cliente)
    -> from prestatario natural join prestamo
    -> group by nombre_sucursal;
Query OK, 0 rows affected (0.026 sec)

MariaDB [banco]> select * from sucursales_prestatario;
+----------------+-----------------------+
| nombre_sucursal | count(nombre_cliente) |
+----------------+-----------------------+
| Becerril        |                     1 |
| Centro          |                     2 |
| Collado Mediano |                     1 |
| Moralzarzal     |                     1 |
| Navacerrada     |                     2 |
+----------------+-----------------------+
5 rows in set (0.008 sec)
```

Lenguaje de Consultas (SQL)

❖ Usuarios y autorización

Hemos realizado todas las actividades en MySQL como usuario root con acceso a todas las bases de datos. Sin embargo, en muchos casos se requieren más restricciones, existen formas de crear usuarios con permisos personalizados.
Para crear un nuevo usuario en el shell de MySQL:

```
CREATE USER 'newuser'@'localhost' IDENTIFIED BY 'password';
```

Lenguaje de Consultas (SQL)

Usuarios y autorización

- Se pueden asignar a los usuarios varios tipos de autorización para diferentes partes de la base de datos.

Por ejemplo:

- La autorización de lectura
- La autorización de inserción
- La autorización de actualización
- La autorización de borrado

Cada uno de estos tipos de autorización se denomina privilegio.

Lenguaje de Consultas (SQL)

Usuarios y autorización

- Se puede conceder a cada usuario todos los tipos de privilegios, ninguno de ellos o una combinación de los mismos sobre partes concretas de la base de datos, como puede ser una relación o una vista.
- SQL incluye los privilegios **select**, **insert**, **update** y **delete**.
- Y los privilegios de crear, borrar o modificar relaciones y ejecutar procedimientos.
- **All privileges** puede utilizarse como forma abreviada de todos los privilegios que se pueden conceder. El usuario que crea una relación nueva recibe de manera automática todos los privilegios sobre esa relación.

Lenguaje de Consultas (SQL)

❖ Usuarios y autorización

¿Cómo se conceden y retiran los privilegios en SQL?

La instrucción **grant** se utiliza para conceder autorizaciones.

grant <lista de privilegios> **on** <nombre de relación o de vista> **to** <lista de usuarios o de roles>

Por ejemplo:

grant select on *cuenta* **to** *Martín, María*

Concede a los usuarios **Martín** y **María** la autorización **select** sobre la relación **cuenta**.

Lenguaje de Consultas (SQL)

Usuarios y autorización

La autorización **update** puede concederse sobre todos los atributos de la relación o sólo sobre algunos.

grant update (*importe*) **on** *préstamo* **to** *Martín, María*

Concede a los usuarios Martín y María la autorización **update** sobre el atributo *importe* de la relación *préstamo*.

Si se omite la lista de atributos, el privilegio update se concede sobre todos los atributos de la relación.

Lenguaje de Consultas (SQL)

Usuarios y autorización

Para retirar una autorización se emplea la instrucción **revoke**.

revoke <lista de privilegios> **on** <nombre de la relación o nombre de la vista>
from <lista de usuarios o de roles>

Para retirar los privilegios concedidos previamente:

revoke select on *sucursal* **from** *Martín, María*
revoke update (*importe*) **on** *préstamo* **from** *Martín, María*

Lenguaje de Consultas (SQL)

- **Usuarios y autorización**

Práctica

Lenguaje de Consultas (SQL)

Usuarios y autorización

Nuestro usuario root no tiene password, para agregarlo:

```
mysql> ALTER USER 'root'@'localhost' IDENTIFIED BY '1234';
mysql> flush privileges;
```

En adelante el logueo a mysql será:

```
Mysql –u root –p
Enter password:
```

Lenguaje de Consultas (SQL)

Usuarios y autorización

Algunos ejemplos:

```
GRANT ALL PRIVILEGES ON *.* TO 'nombre_usuario'@'localhost';
```

Estamos otorgando a un usuario todos los permisos sobre todas las bases de datos.

Los asteriscos indican que los permisos serán asignados a todas las bases de datos y a todas las tablas (primer asterisco todas bases de datos, segundo asterisco todas las tablas).

Lenguaje de Consultas (SQL)

❖ Usuarios y autorización

CREATE USER 'newuser'@'localhost' IDENTIFIED BY 'password';

```
MariaDB [(none)]> create user 'banco'@'localhost' identified by 'banco';
Query OK, 0 rows affected (0.00 sec)
```

```
MariaDB [(none)]> GRANT SELECT, INSERT, UPDATE, DELETE, CREATE, DROP
    -> ON banco.*
    -> TO 'banco'@'localhost';
Query OK, 0 rows affected (0.01 sec)
```

```
MariaDB [(none)]> flush privileges;
Query OK, 0 rows affected (0.01 sec)
```

El nuevo usuario ***banco*** podrá consultar, crear, actualizar y eliminar registros, así cómo podrá crear o eliminar elementos (tablas, índices, columnas, etc.) y todos estos permisos serán válidos únicamente en la base de datos "**banco**" y se aplicarán a todas las tablas.

Lenguaje de Consultas (SQL)

Usuarios y autorización

```
MariaDB [(none)]> quit
Bye

Mamá@DESKTOP-JT3175H c:\xampp
# mysql -u banco -p
Enter password: *****
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 13
Server version: 10.4.21-MariaDB mariadb.org binary distribution

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MariaDB [(none)]> use entidadbancaria;
ERROR 1044 (42000): Access denied for user 'banco'@'localhost' to database 'entidadbancaria'
MariaDB [(none)]>
```

Se le ha denegado el acceso al usuario ‘banco’ a la base de datos ‘entidad bancaria’

Lenguaje de Consultas (SQL)

Usuarios y autorización

```
MariaDB [(none)]> REVOKE CREATE, DROP ON banco.* from 'banco'@'localhost';
Query OK, 0 rows affected (0.00 sec)
```

```
MariaDB [(none)]> flush privileges;
Query OK, 0 rows affected (0.01 sec)
```

Removemos los permisos de crear y eliminar elementos (tablas, índices, columnas, etc.) al usuario ***banco*** de la base de datos "**banco**".

Lenguaje de Consultas (SQL)

Usuarios y autorización

```
MariaDB [banco]> create user 'mercadotecnia'@'localhost' identified by 'merca';
Query OK, 0 rows affected (0.006 sec)

MariaDB [banco]> flush privileges;
Query OK, 0 rows affected (0.001 sec)
```

Agregamos un usuario ‘mercadotecnia’ al que le daremos permisos para acceder a las vistas creadas.

Lenguaje de Consultas (SQL)

Usuarios y autorización

```
MariaDB [banco]> GRANT SELECT
    -> ON banco.sucursales_cuenta
    -> TO 'mercadotecnia'@'localhost';
Query OK, 0 rows affected (0.007 sec)

MariaDB [banco]> flush privileges;
Query OK, 0 rows affected (0.002 sec)
```

Otorgamos permisos al usuario ‘mercadotecnia’ para la vista ‘sucursales_cuenta’.

Lenguaje de Consultas (SQL)

Usuarios y autorización

Lista de permisos:

ALL PRIVILEGES: Acceso completo a una base de datos.

CREATE: Permite crear nuevas tablas o bases de datos.

DROP: Permite eliminar tablas o bases de datos.

DELETE: Permite eliminar filas de las tablas.

INSERT: Permite insertar filas en las tablas.

SELECT: Permite usar el comando SELECT para leer las bases de datos.

UPDATE: Permite actualizar las filas de las tablas.

Lenguaje de Consultas (SQL)

Usuarios y autorización

Otros comandos

Mostrar los permisos actuales de un usuario:

SHOW GRANTS FOR 'username'@'localhost’;

Eliminar un usuario:

DROP USER 'username'@'localhost';

Listado de todos los usuarios:

SELECT User FROM mysql.user;

Lenguaje de Consultas (SQL)

Transacciones

- **Transacción** hace referencia a un conjunto de operaciones que forman una única unidad lógica de trabajo.
- Por ejemplo, la transferencia de dinero de una cuenta a otra es una transacción que consta de dos actualizaciones al menos, una para cada cuenta.
- Resulta importante que, o bien se ejecuten completamente todas las acciones de una transacción, o bien, en caso de fallo, se deshagan los efectos parciales de cada transacción incompleta. Esta propiedad se denomina **atomicidad**.

Lenguaje de Consultas (SQL)

Transacciones

- Una vez ejecutada con éxito una transacción, sus efectos deben persistir en la base de datos
- Un fallo en el sistema no debe tener como consecuencia que la base de datos descarte una transacción que se haya completado con éxito.
- Esta propiedad se denomina **durabilidad**.

Lenguaje de Consultas (SQL)

Transacciones

- Cuando varias transacciones se ejecutan de manera concurrente, si no se controlan las actualizaciones de los datos compartidos, existe la posibilidad de que las transacciones operen sobre estados intermedios inconsistentes creados por las actualizaciones de otras transacciones.
- Los sistemas de bases de datos deben proporcionar los mecanismos para aislar las transacciones de otras transacciones que se ejecuten de manera concurrente. Esta propiedad se denomina **aislamiento**.

Lenguaje de Consultas (SQL)

Transacciones

- **Consistencia**. La ejecución aislada de la transacción (es decir, sin otra transacción que se ejecute concurrentemente) conserva la consistencia de la base de datos.
- Estas propiedades a menudo reciben el nombre de propiedades ACID (Atomicity, Consistency, Isolation y Durability).
- Para asegurar la integridad de los datos se necesita que el sistema de base de datos mantenga estas propiedades de las transacciones.

Lenguaje de Consultas (SQL)

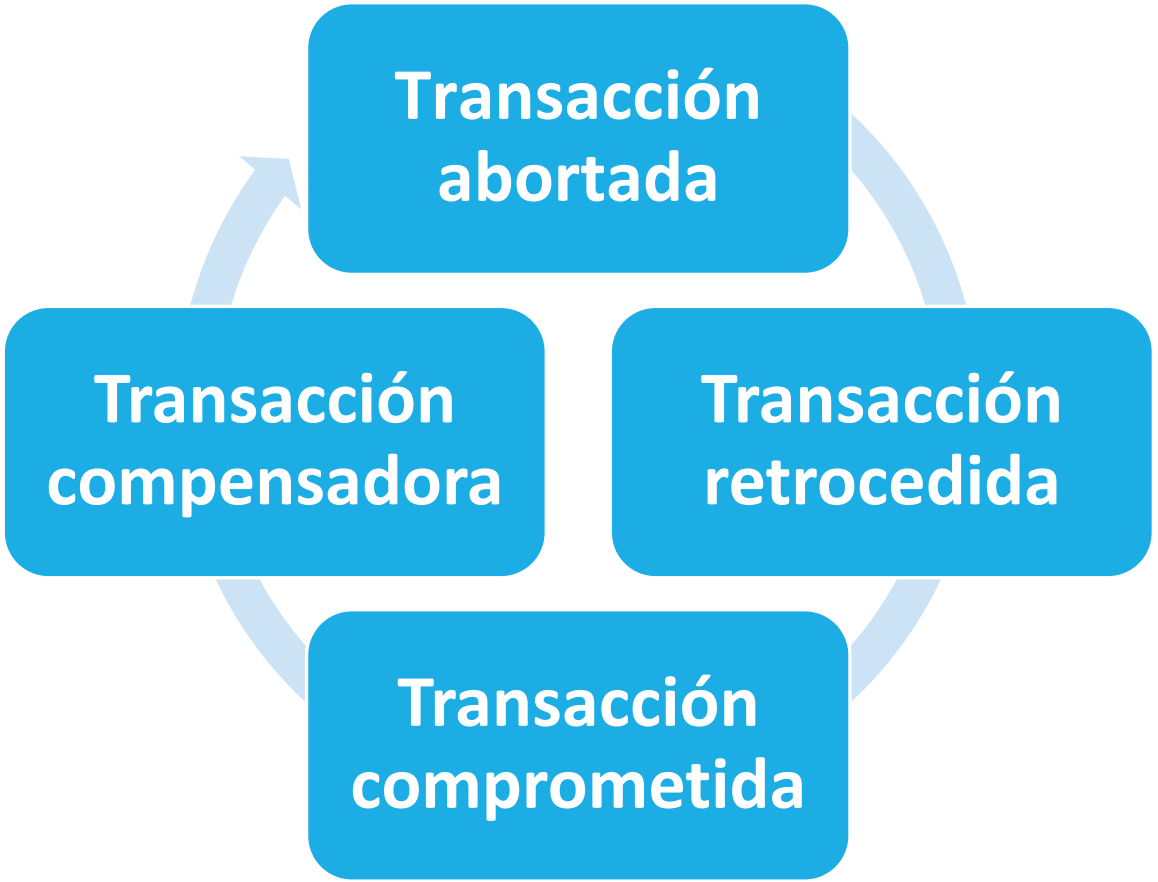
Transacciones

- Una transacción se inicia por la ejecución de un programa de usuario escrito en un lenguaje de manipulación de datos de alto nivel o en un lenguaje de programación (por ejemplo SQL,C++ o Java)
- Está delimitado por instrucciones **begin transaction** (inicio transacción) y **end transaction** (fin transacción).
- La transacción consiste en todas las operaciones que se ejecutan entre begin transaction y end transaction.

Lenguaje de Consultas (SQL)

Transacciones

- **Estados de una transacción:**



Lenguaje de Consultas (SQL)

❖ Transacciones

- **Estados de una transacción:**

✓ **Transacción abortada.** Transacción que no termina su ejecución con éxito.

✓ **Transacción retrocedida**. Cualquier cambio que haya hecho la transacción abortada sobre la base de datos debe deshacerse. Una vez que se han deshecho los cambios efectuados por la transacción abortada, se dice que la transacción está retrocedida.

Lenguaje de Consultas (SQL)

❖ Transacciones

- **Estados de una transacción:**

✓**Transacción comprometida.** Una transacción que termina con éxito.

✓**Transacción compensadora**. La única forma de deshacer los cambios de una transacción comprometida es ejecutando una transacción compensadora.

Lenguaje de Consultas (SQL)

Transacciones

- Transacción comprometida vs transacción compensadora

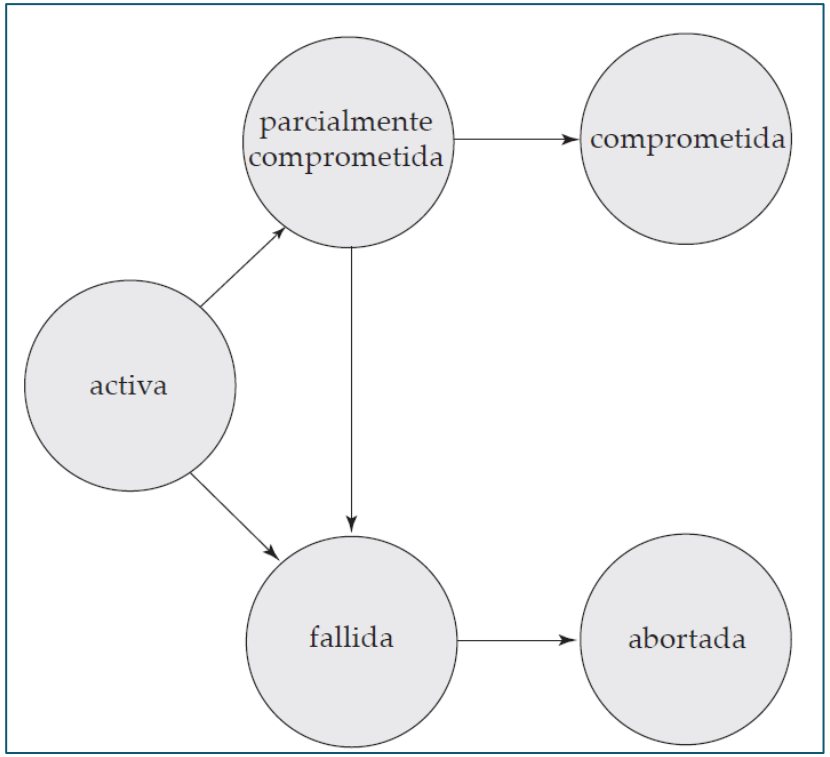
Por ejemplo, si una transacción añade $20 a una cuenta, la transacción compensadora debería restar $20 de la cuenta.

Lenguaje de Consultas (SQL)

Transacciones

Una transacción debe estar en uno de los estados siguientes:

- **Activa**, el estado inicial; la transacción permanece en este estado durante su ejecución.
- **Parcialmente comprometida**, después de ejecutarse la última instrucción.
- **Fallida**, tras descubrir que no puede continuar la ejecución normal.
- **Abortada**, después del retroceso de la transacción y de haber restablecido la base de datos a su estado anterior al comienzo de la transacción.
- **Comprometida**, tras completarse con éxito.



Lenguaje de Consultas (SQL)

Transacciones

- Una transacción llega al estado fallida después de que el sistema determine que dicha transacción no puede continuar su ejecución normal (por ejemplo, a causa de errores de hardware o lógicos).
- Una transacción de este tipo se debe retroceder. Después pasa al estado abortada. En este punto, el sistema tiene dos opciones:

Lenguaje de Consultas (SQL)

Transacciones

- **Reiniciar la transacción**, pero sólo si la transacción se ha abortado a causa de algún error hardware o software que no lo haya provocado la lógica interna de la transacción. Una transacción reiniciada se considera una nueva transacción.
- **Cancelar la transacción.** Normalmente se hace esto si hay algún error interno lógico que sólo se puede corregir escribiendo de nuevo el programa de aplicación, o debido a una entrada incorrecta o a que no se han encontrado los datos deseados en la base de datos.

Lenguaje de Consultas (SQL)

Transacciones

- La transacción T1 transfiere 50 de la cuenta A a la cuenta B.
- La transacción T2 transfiere el 10 por ciento del saldo de la cuenta A a la cuenta B, y se define:

T_i: leer(A);
$A := A - 50$;
escribir(A);
leer(B);
$B := B + 50$;
escribir(B).

T_2: leer(A);
$temp := A * 0.1$;
$A := A - temp$;
escribir(A);
leer(B);
$B := B + temp$;
escribir(B).

Lenguaje de Consultas (SQL)

Transacciones

- En el ejemplo común T1, en el que una cantidad de dinero es transferida de la cuenta de una persona a otra y se requerirían por lo menos dos consultas:

```
UPDATE cuentas SET balance = balance - cantidad_transferida WHERE cliente = persona1;
```

```
UPDATE cuentas SET balance = balance + cantidad_transferida WHERE cliente = persona2;
```

¿Qué sucedería si algo inesperado ocurre a mitad de la transacción y se cae el sistema?

La persona1 pensaría que ha realizado la transferencia, pues el dinero ya no está en su cuenta.

Y la persona2 estaría molesta pues no recibió la transferencia.

Lenguaje de Consultas (SQL)

- **Transacciones**
  - En este ejemplo tan sencillo se ilustra la necesidad de que estas consultas sean ejecutadas de manera conjunta, o en su caso, que no se ejecute ninguna de ellas. Es aquí donde las transacciones toman un papel muy importante.

Lenguaje de Consultas (SQL)

Transacciones

- Los pasos para usar transacciones en MySQL son:
  1) Iniciar una transacción con el uso de la sentencia BEGIN.
  2) Actualizar, insertar o eliminar registros en la base de datos.
  3) Si se quieren los cambios a la base de datos, completar la transacción con el uso de la sentencia COMMIT. Únicamente cuando se procesa un COMMIT los cambios hechos por las consultas serán permanentes.
  4) Si sucede algún problema, podemos hacer uso de la sentencia ROLLBACK para cancelar los cambios que han sido realizados por las consultas que han sido ejecutadas hasta el momento.

Lenguaje de Consultas (SQL)

Transacciones

```
MariaDB [(none)]> use banco;
Database changed
MariaDB [banco]> show columns from sucursal;
+-----------------+--------------+------+-----+---------+-------+
| Field           | Type         | Null | Key | Default | Extra |
+-----------------+--------------+------+-----+---------+-------+
| nombre_sucursal | varchar(100) | NO   | PRI | NULL    |       |
| ciudad_sucursal | varchar(255) | NO   |     | NULL    |       |
| activos         | int(20)      | NO   |     | NULL    |       |
+-----------------+--------------+------+-----+---------+-------+
3 rows in set (0.02 sec)
```

Transacciones

- Transacción que inserta una tupla en la relación **sucursal**.

```
MariaDB [banco]> BEGIN;
Query OK, 0 rows affected (0.00 sec)

MariaDB [banco]> INSERT INTO `sucursal` (`nombre_sucursal`, `ciudad_sucursal`, `activos`) VALUES ('Animas','Xalapa','500000');
Query OK, 1 row affected (0.00 sec)

MariaDB [banco]> select * from sucursal;
+---------------------+-------------------+---------+
| nombre_sucursal     | ciudad_sucursal   | activos |
+---------------------+-------------------+---------+
| Animas              | Xalapa            |  500000 |
| Becerril            | Aluche            |  400000 |
| Centro              | Arganzuela        | 9000000 |
| Collado Mediano     | Aluche            | 8000000 |
| Galapagar           | Arganzuela        | 7100000 |
| Moralzarzal         | La Granja         | 2100000 |
| Navacerrada         | Aluche            | 1700000 |
| Navas de la Asunción | Alcalá de Henares |  300000 |
| Segovia             | Cerceda           | 3700000 |
+---------------------+-------------------+---------+
9 rows in set (0.00 sec)
```

Lenguaje de Consultas (SQL)

Transacciones

- Con la instrucción ROLLBACK deshacemos la operación realizada.

```
MariaDB [banco]> ROLLBACK;
Query OK, 0 rows affected (0.00 sec)

MariaDB [banco]> select * from sucursal;
+----------------------+--------------------+---------+
| nombre_sucursal      | ciudad_sucursal    | activos |
+----------------------+--------------------+---------+
| Becerril             | Aluche             |  400000 |
| Centro               | Arganzuela         | 9000000 |
| Collado Mediano      | Aluche             | 8000000 |
| Galapagar            | Arganzuela         | 7100000 |
| Moralzarzal          | La Granja          | 2100000 |
| Navacerrada          | Aluche             | 1700000 |
| Navas de la Asunción | Alcalá de Henares |  300000 |
| Segovia              | Cerceda            | 3700000 |
+----------------------+--------------------+---------+
8 rows in set (0.00 sec)
```

❖Transacciones

- si perdemos la conexión al servidor antes de que la transacción sea completada con la instrucción COMMIT, las operaciones no son aplicadas.

```
MariaDB [banco]> BEGIN;
Query OK, 0 rows affected (0.00 sec)

MariaDB [banco]> INSERT INTO `sucursal` (`nombre_sucursal`, `ciudad_sucursal`, `activos`) VALUES ('Animas','Xalapa','500000');
Query OK, 1 row affected (0.00 sec)

MariaDB [banco]> select * from sucursal;
+----------------------+-------------------+---------+
| nombre_sucursal      | ciudad_sucursal   | activos |
+----------------------+-------------------+---------+
| Animas               | Xalapa            |  500000 |
| Becerril             | Aluche            |  400000 |
| Centro               | Arganzuela        | 9000000 |
| Collado Mediano      | Aluche            | 8000000 |
| Galapagar            | Arganzuela        | 7100000 |
| Moralzarzal          | La Granja         | 2100000 |
| Navacerrada          | Aluche            | 1700000 |
| Navas de la Asunción | Alcalá de Henares |  300000 |
| Segovia              | Cerceda           | 3700000 |
+----------------------+-------------------+---------+
9 rows in set (0.00 sec)

MariaDB [banco]> quit;
Bye
```

Lenguaje de Consultas (SQL)

```
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 67
Server version: 10.1.38-MariaDB mariadb.org binary distribution

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MariaDB [(none)]> use banco;
Database changed
MariaDB [banco]> select * from sucursal;
+----------------------+------------------+---------+
| nombre_sucursal      | ciudad_sucursal  | activos |
+----------------------+------------------+---------+
| Becerril             | Aluche           |  400000 |
| Centro               | Arganzuela       | 9000000 |
| Collado Mediano      | Aluche           | 8000000 |
| Galapagar            | Arganzuela       | 7100000 |
| Moralzarzal          | La Granja        | 2100000 |
| Navacerrada          | Aluche           | 1700000 |
| Navas de la Asunción | Alcalá de Henares |  300000 |
| Segovia              | Cerceda          | 3700000 |
+----------------------+------------------+---------+
8 rows in set (0.00 sec)
```

Lenguaje de Consultas (SQL)

Transacciones

- Si terminamos la transacción con las instrucción COMMIT y perdemos la conexión, los cambios perduran.

```
MariaDB [banco]> BEGIN;
Query OK, 0 rows affected (0.00 sec)

MariaDB [banco]> INSERT INTO `sucursal` (`nombre_sucursal`, `ciudad_sucursal`, `activos`) VALUES ('Animas','Xalapa','500000');
Query OK, 1 row affected (0.00 sec)

MariaDB [banco]> COMMIT;
Query OK, 0 rows affected (0.00 sec)

MariaDB [banco]> QUIT;
Bye
```

```
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 68
Server version: 10.1.38-MariaDB mariadb.org binary distribution

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MariaDB [(none)]> use banco;
Database changed
MariaDB [banco]> select * from sucursal;
+----------------------+--------------------+---------+
| nombre_sucursal      | ciudad_sucursal    | activos |
+----------------------+--------------------+---------+
| Animas               | Xalapa             |  500000 |
| Becerril             | Aluche             |  400000 |
| Centro               | Arganzuela         | 9000000 |
| Collado Mediano      | Aluche             | 8000000 |
| Galapagar            | Arganzuela         | 7100000 |
| Moralzarzal          | La Granja          | 2100000 |
| Navacerrada          | Aluche             | 1700000 |
| Navas de la Asunción | Alcalá de Henares |  300000 |
| Segovia              | Cerceda            | 3700000 |
+----------------------+--------------------+---------+
9 rows in set (0.00 sec)
```

Transacciones

- **Lecturas consistentes.** Cuando una sentencia SELECT es ejecutada, MySQL regresa los valores presentes en la base de datos hasta la transacción más reciente que ha sido completada.

```
MariaDB [banco]> select * from cuenta;
+--------------+----------------+-------+
| numero_cuenta | nombre_sucursal | saldo |
+--------------+----------------+-------+
| C-101         | Centro         |   500 |
| C-102         | Navacerrada    |   400 |
| C-201         | Galapagar      |   900 |
| C-215         | Becerril       |   700 |
| C-217         | Galapagar      |   750 |
| C-222         | Moralzarzal    |   700 |
| C-305         | Collado Mediano |   350 |
+--------------+----------------+-------+
7 rows in set (0.00 sec)
```

Si alguna transacción está en progreso, los cambios hechos por alguna sentencia INSERT o UPDATE no serán reflejados. **Solo las transacciones abiertas si pueden ver sus propios cambios.**

```
MariaDB [banco]> BEGIN;
Query OK, 0 rows affected (0.00 sec)
```

```
MariaDB [banco]> UPDATE cuenta SET saldo = '950' WHERE numero_cuenta = 'C-101';
Query OK, 1 row affected (0.00 sec)
Rows matched: 1  Changed: 1  Warnings: 0
```

XAMPP for Windows - mysql -u root -p

```
MariaDB [banco]> UPDATE cuenta SET saldo = '950' WHERE numero_cuenta = 'C-101';
Query OK, 1 row affected (0.00 sec)
Rows matched: 1  Changed: 1  Warnings: 0

MariaDB [banco]> select * from cuenta;
+---------------+-----------------+-------+
| numero_cuenta | nombre_sucursal | saldo |
+---------------+-----------------+-------+
| C-101         | Centro          |   950 |
| C-102         | Navacerrada     |   400 |
| C-201         | Galapagar       |   900 |
| C-215         | Becerril        |   700 |
| C-217         | Galapagar       |   750 |
| C-222         | Moralzarzal     |   700 |
| C-305         | Collado Mediano |   350 |
+---------------+-----------------+-------+
7 rows in set (0.01 sec)
```

XAMPP for Windows - mysql -u root -p

```
Setting environment for using XAMPP for Windows.
erika@DESKTOP-34SETHG c:\xampp
# mysql -u root -p
Enter password: ****
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 69
Server version: 10.1.38-MariaDB mariadb.org binary distribution

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MariaDB [(none)]> use banco;
Database changed
MariaDB [banco]> select * from cuenta;
+---------------+-----------------+-------+
| numero_cuenta | nombre_sucursal | saldo |
+---------------+-----------------+-------+
| C-101         | Centro          |   600 |
| C-102         | Navacerrada     |   400 |
| C-201         | Galapagar       |   900 |
| C-215         | Becerril        |   700 |
| C-217         | Galapagar       |   750 |
| C-222         | Moralzarzal     |   700 |
| C-305         | Collado Mediano |   350 |
+---------------+-----------------+-------+
7 rows in set (0.01 sec)

MariaDB [banco]>
```

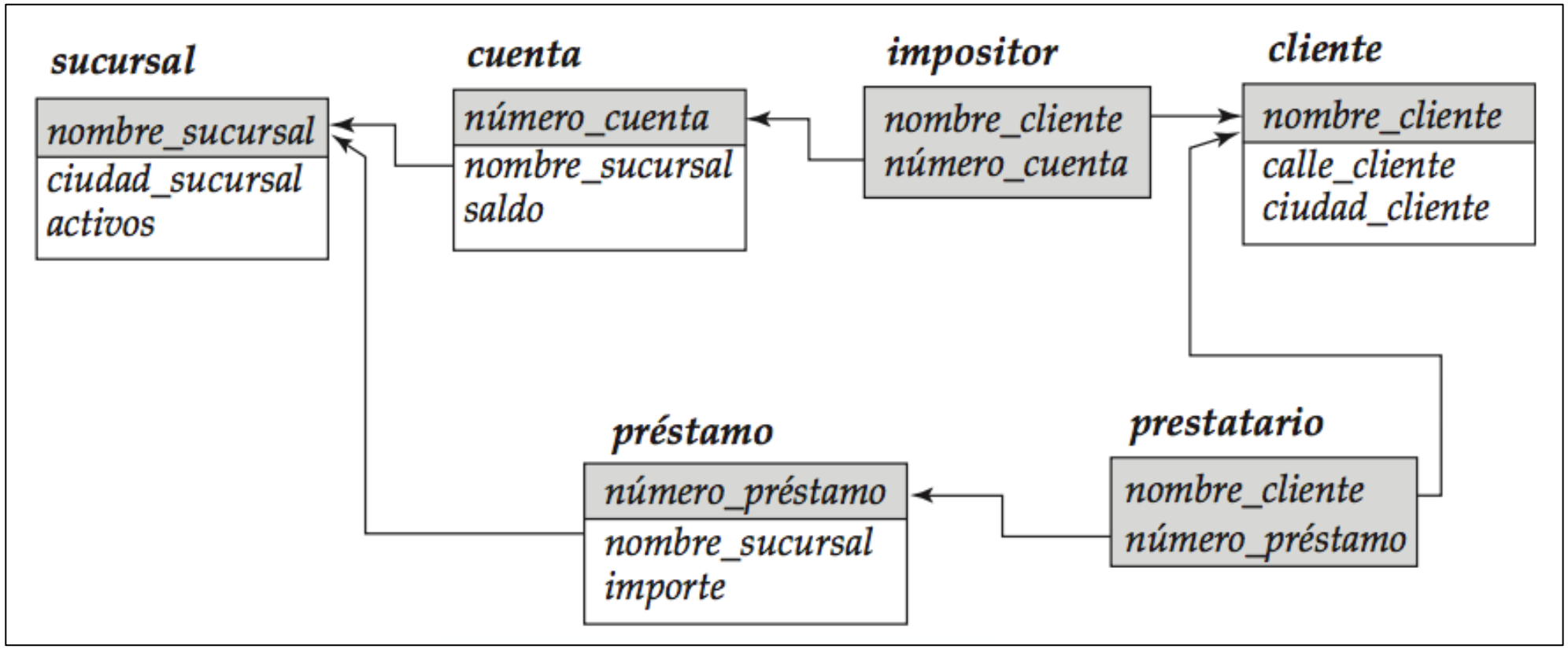
Lenguaje de Consultas (SQL)

XAMPP for Windows - mysql -u root -p

```
MariaDB [banco]> select * from cuenta;
+--------------+-----------------+-------+
| numero_cuenta | nombre_sucursal | saldo |
+--------------+-----------------+-------+
| C-101        | Centro          |   950 |
| C-102        | Navacerrada     |   400 |
| C-201        | Galapagar       |   900 |
| C-215        | Becerril        |   700 |
| C-217        | Galapagar       |   750 |
| C-222        | Moralzarzal     |   700 |
| C-305        | Collado Mediano |   350 |
+--------------+-----------------+-------+
7 rows in set (0.01 sec)

MariaDB [banco]> COMMIT;
Query OK, 0 rows affected (0.01 sec)

MariaDB [banco]>
```

XAMPP for Windows - mysql -u root -p

```
MariaDB [banco]> select * from cuenta;
+--------------+-----------------+-------+
| numero_cuenta | nombre_sucursal | saldo |
+--------------+-----------------+-------+
| C-101        | Centro          |   950 |
| C-102        | Navacerrada     |   400 |
| C-201        | Galapagar       |   900 |
| C-215        | Becerril        |   700 |
| C-217        | Galapagar       |   750 |
| C-222        | Moralzarzal     |   700 |
| C-305        | Collado Mediano |   350 |
+--------------+-----------------+-------+
7 rows in set (0.00 sec)

MariaDB [banco]>
```


sucursal
nombre_sucursal
ciudad_sucursal
activos
cuenta
número_cuenta
nombre_sucursal
saldo
impositor
nombre_cliente
número_cuenta
cliente
nombre_cliente
calle_cliente
ciudad_cliente
préstamo
número_préstamo
nombre_sucursal
importe
prestatario
nombre_cliente
número_préstamo

Lenguaje de Consultas (SQL)

Transacciones

- **Actividad 1:**
- Realiza una transacción sobre la base de datos Banco que:

1) Agregue un nuevo cliente en la tabla CLIENTE con nombre de cliente **Salas**.

2) Agregar un Préstamo en la tabla PRESTAMO con número de cuenta **P-35.**

3) Agregar en la tabla PRESTATARIO la cuenta **P-35** al cliente **Salas**.

4) Actualizar la tabla SUCURSAL reduciendo el activo con la cantidad otorgada en el préstamo.

```
BEGIN;

INSERT INTO cliente (nombre_cliente,calle_cliente,ciudad_cliente) VALUES ('Salas',
'Lucio', 'Cerceda’);

INSERT INTO prestamo (numero_prestamo, nombre_sucursal, importe) VALUES
('P-35', 'Collado Mediano', 1000000);

INSERT INTO prestatario (nombre_cliente, numero_prestamo) VALUES
('Salas', 'P-35’);

UPDATE sucursal SET activos=activos-1000000 WHERE nombre_sucursal="Collado Mediano";

COMMIT;
```

Lenguaje de Consultas (SQL)

Transacciones

- **Actividad 2:**
- Realiza una transacción sobre la base de datos Banco que:

1) Agregue un nuevo cliente en la tabla CLIENTE con nombre de cliente **Guerrero**.

2) Agregar una cuenta en la tabla CUENTA con número de cuenta **C-403.**

3) Asignar en la tabla IMPOSITOR la cuenta C-403 al cliente Guerrero.

4) Agrega en la tabla SUCURSAL el saldo de la cuenta recién creada al atributo *activos*.

Lenguaje de Consultas (SQL)

Gracias por su atención